



Detection of faulty Beam Position Monitors

E. Fol, R. Tomas Garcia

Machine Learning Applications for Particle Accelerators, Feb. 28 – March 2, 2018, SLAC National Accelerator Laboratory

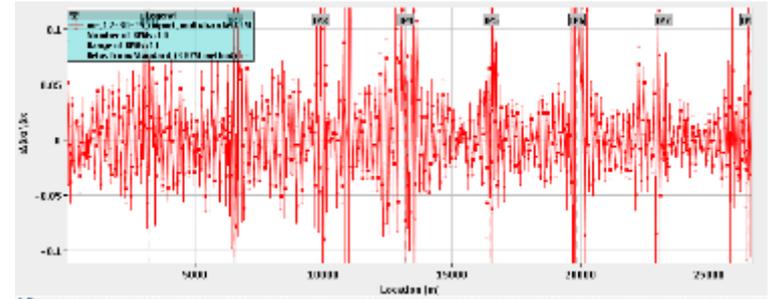
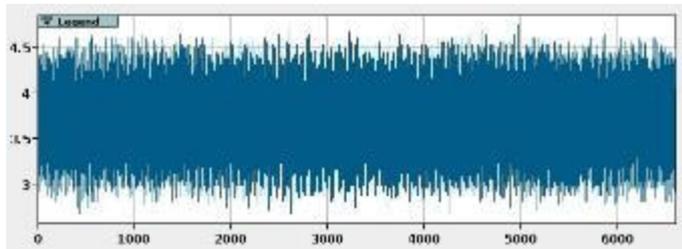


Content

1. Introduction to optics measurements at LHC
2. Faulty BPM data
3. Cluster analysis of measured signal
4. Autoencoder for anomaly detection
5. Conclusion and future tasks

Introduction to optics measurements at LHC

- Beam position monitors (BPMs) record the turn-by-turn data measuring the oscillations of the excited beam
- Reconstruction of the optics mainly concerned by the phase advances obtained from the measured data using Fourier transformation analysis

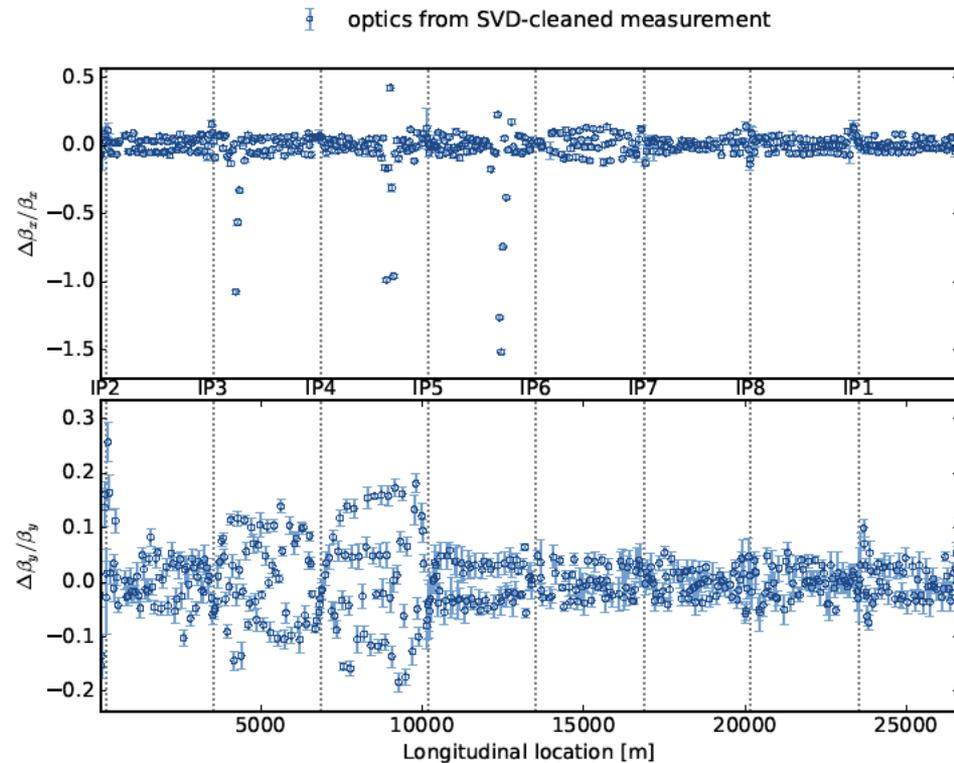
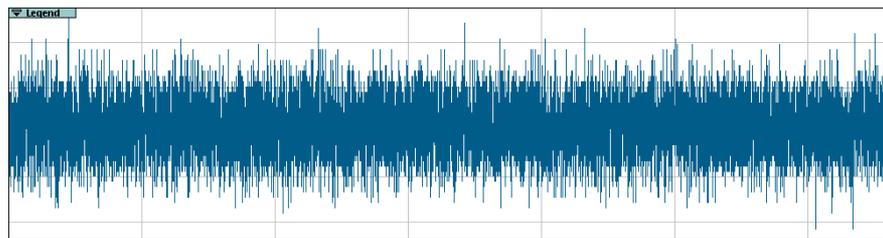
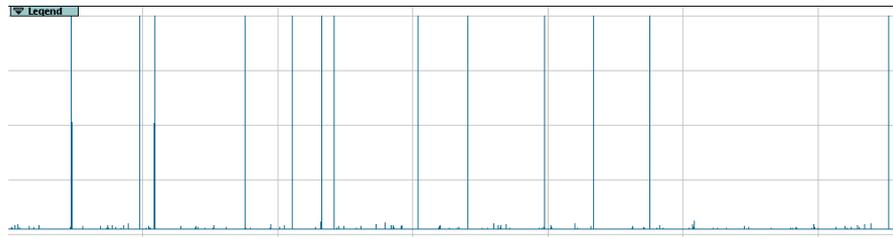


- One of the main parameters is beta-beating: ratio of the measured betatron function with respect to the nominal designed function
- The measured optics is compared then to the nominal model of the machine in order to compute the corrections

1. Faulty BPM data

- 1024 BPMs per beam around the ring to measure the turn-by-turn data
 - Statistical analysis of the past measurements show that ~10% of BPMs are faulty
- Optics functions are affected by occurrences of faulty signal
 - some artifacts cannot be recognized in turn-by-turn data
 - reasons, relevant features and their correlations are unknown
 - time-consuming manual “cleaning” and analysis re-computation
- Unphysical values still escape our current automatic cleaning tools
→ ML as an alternative solution to improve the optics measurements

1. Faulty BPM data



1. Faulty BPM data

- Current cleaning tools
 - Basic pre-defined cuts (distance between signal peaks, flat signals, etc.) and known bad BPMs are removed in the first step
 - Advanced techniques*:
 - SVD: Singular vectors represent the variation of physical modes describing the beam motion: faulty BPMs produce modes with spatial vectors consisting of localized peaks defined by a certain threshold.
 - Fourier transform in windows: faulty signal show a randomly populated Fourier spectrum: computing the RMS of the amplitudes of the spectral lines within one or several windows detect bad BPMs

* R. Calaga and R. Tomas "Statistical analysis of RHIC beam position monitors performance"

2. Cluster analysis of measured signal

Manual cleaning and data observation required before trying to reconstruct the optics from the signals

- doesn't prevent unphysical values in obtained optics functions
- manual cleaning might produce different results
- is likely that some wrong values still escape human intervention

General Idea:

- we do not want to replicate the currently used automatic cleaning tools, so training data set could be built only combining automatic and manual cleaning result
→ **unsupervised learning approach is preferable**
- assuming most of the BPMs measure correctly, the bad BPMs should appear as an anomaly, but the relevant parameters are unknown
- consider all possible parameters and separate into regions
→ **Cluster analysis**

2. Cluster analysis of measured signal

- Signals from faulty BPM can be considered as “outliers” and isolated into a separate cluster
 - Data visualization is needed to choose appropriate clustering method
 - Data preprocessing steps are needed before applying cluster analysis
- How to define algorithm parameters for diverse optics settings?
- How to evaluate the results? What is the relevant parameter?

2. Cluster analysis of measured signal

K-means* as basic clustering approach:

- Starting with randomly chosen data points as cluster centers, each point is assigned to the closest center
- Move centers to the centroids of the clusters, reassign the points to the nearest center
- Repeat until moving the centroid gives no improvement (based on total squared distance between each point and cluster centroid)



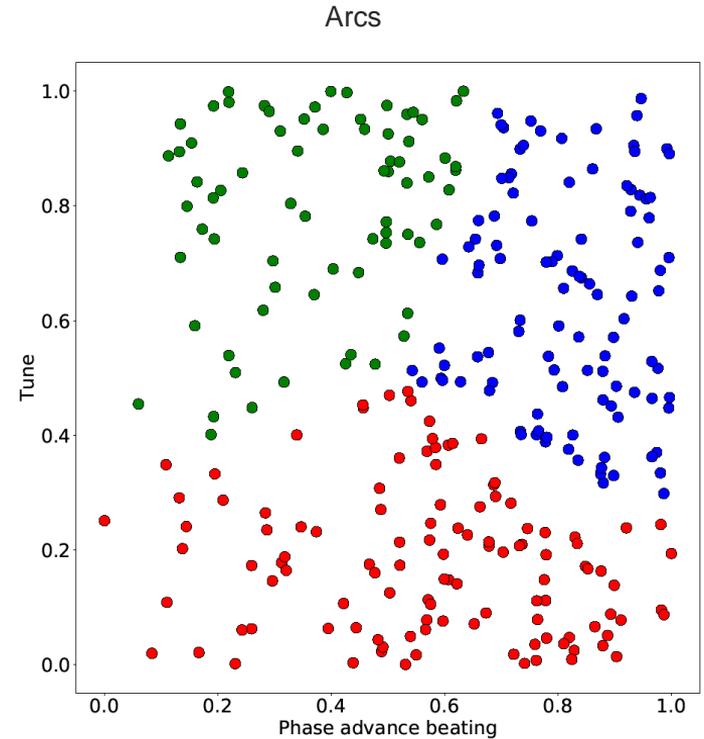
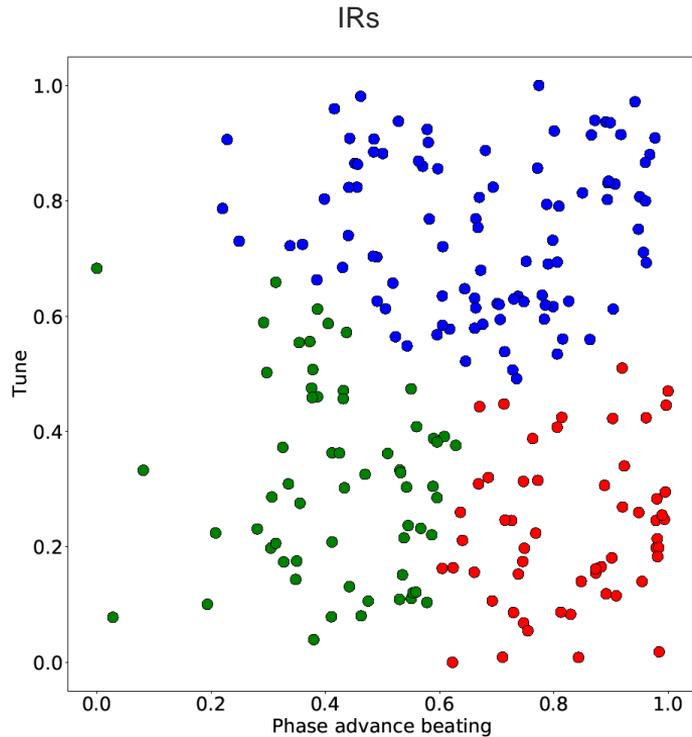
*Stuart P. Lloyd. Least squares quantization in PCM

<http://shabal.in/visuals/kmeans>

2. Cluster analysis of measured signal

- Implementation
 - `sklearn.cluster.KMeans`
 - Features: amplitude, tune, phase advance beating (obtained from harmonic analysis on turn-by-turn data)
 - K-means allows only 2D-Clustering: all possible combinations of features are evaluated
 - all features scaled to the $[0,1]$ range to be represented in a common space
 - IRs and arcs data treated separately since the beam behaves differently in these regions.

K-means with 3 clusters: Optics measurement from LHC Machine Development



2. Cluster analysis of measured signal

- Evaluation of K-means algorithm:
 - Appropriate number of clusters unknown
 - Very sensitive to outliers
 - Choice of the centroids affects the clustering results
 - also an outlier can become a centroid
 - unstable results
- DBSCAN: Density-Based Spatial Clustering of application with noise*

* "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise" Ester, M., H. P. Kriegel, J. Sander, and X. Xu

2. Cluster analysis of measured signal

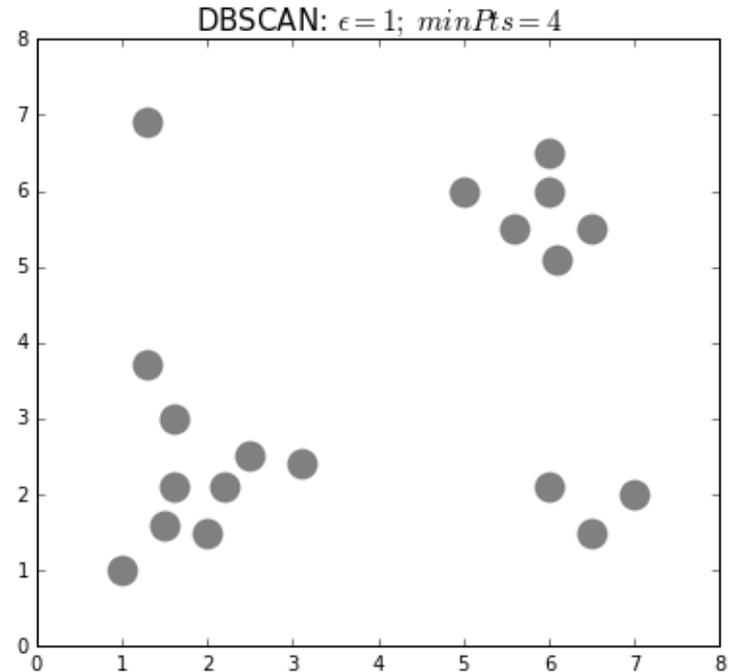
DBSCAN:

- relies on density-based notion of clusters of arbitrary shape
- Core points: build a “neighbourhood” according to a given distance

Non-core points: are inside a neighbourhood, but don't have enough points around to build a new neighbourhood

Noise: don't have enough points in the neighbourhood and are too far away from any of core points

- noise can be easily detected as data points not belonging to any of clusters

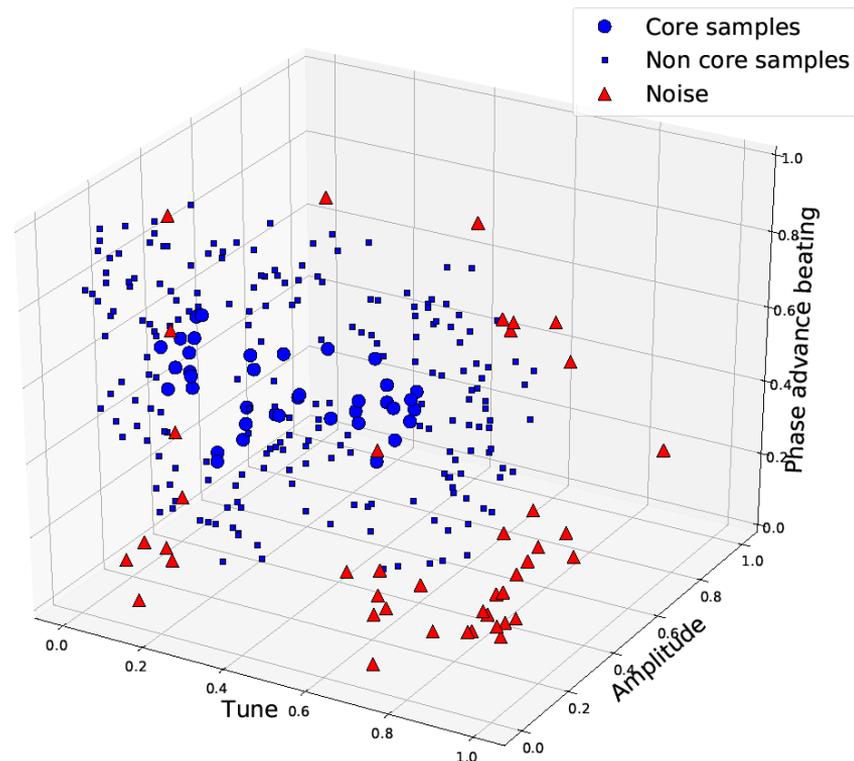


<https://dashee87.github.io/data%20science/general/Clustering-with-Scikit-with-GIFs/>

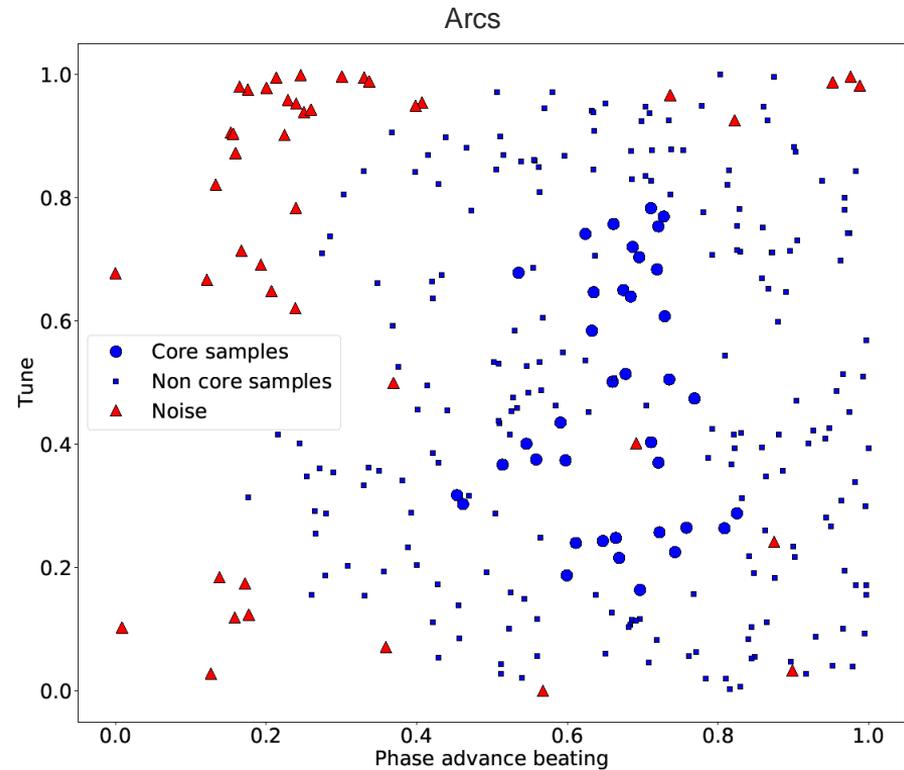
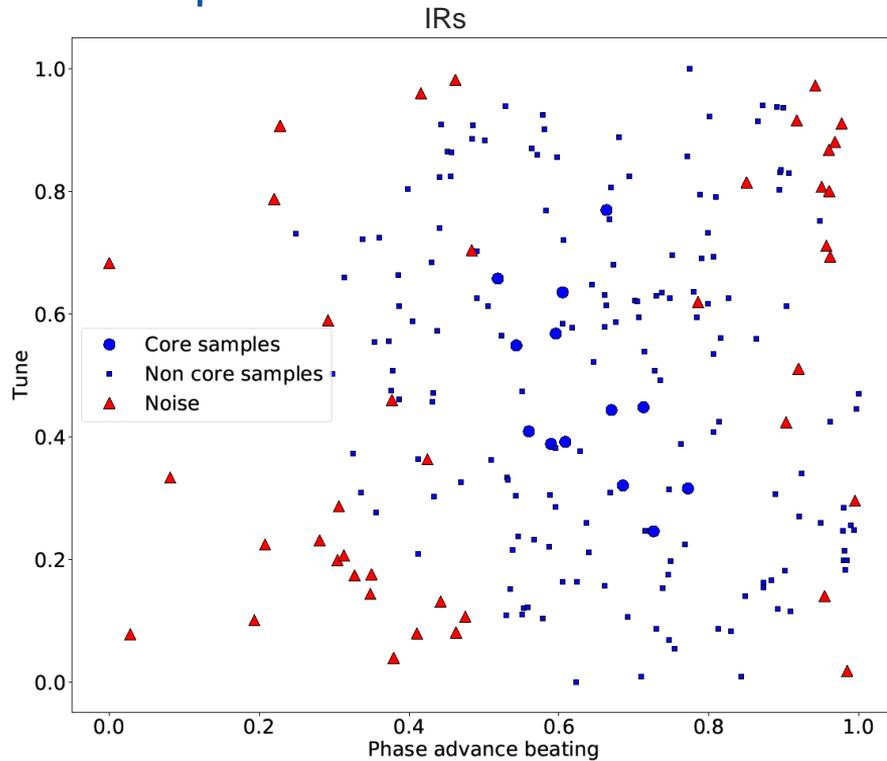
DBSCAN $\epsilon = 0.3$, MinPts = 80, Optics measurement from LHC Machine Development

Implementation:

- Scikit Learn
- tune, amplitude and phase advance beating as features rescaled to [0,1] range
- Clustering in n-dimensional space is possible: no combinations of features are needed
- Set of measurements: we classify a BPM as bad if faulty signal is detected in at least one measurement



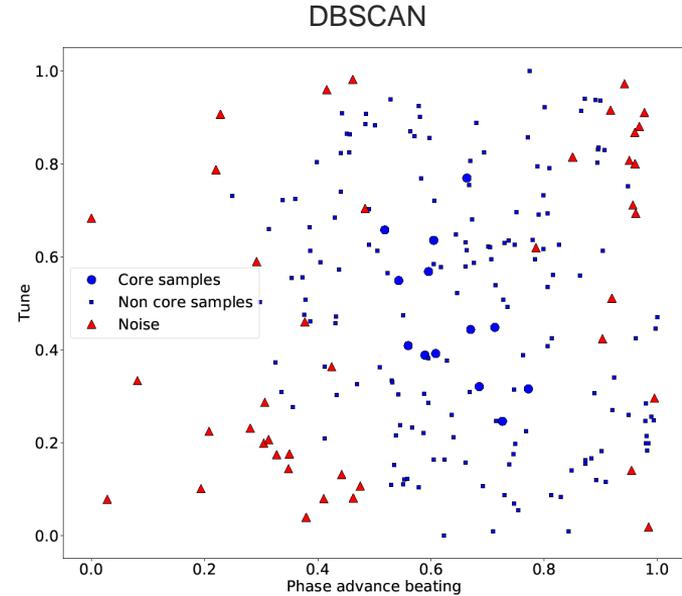
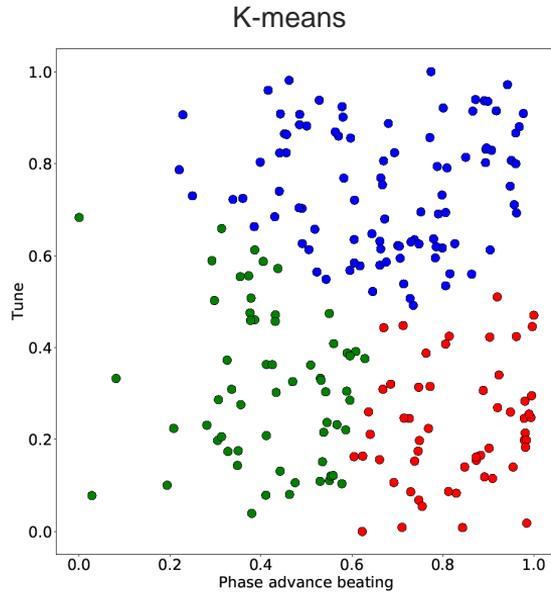
DBSCAN $\epsilon = 0.3$, MinPts = 80, Optics measurement from LHC Machine Development



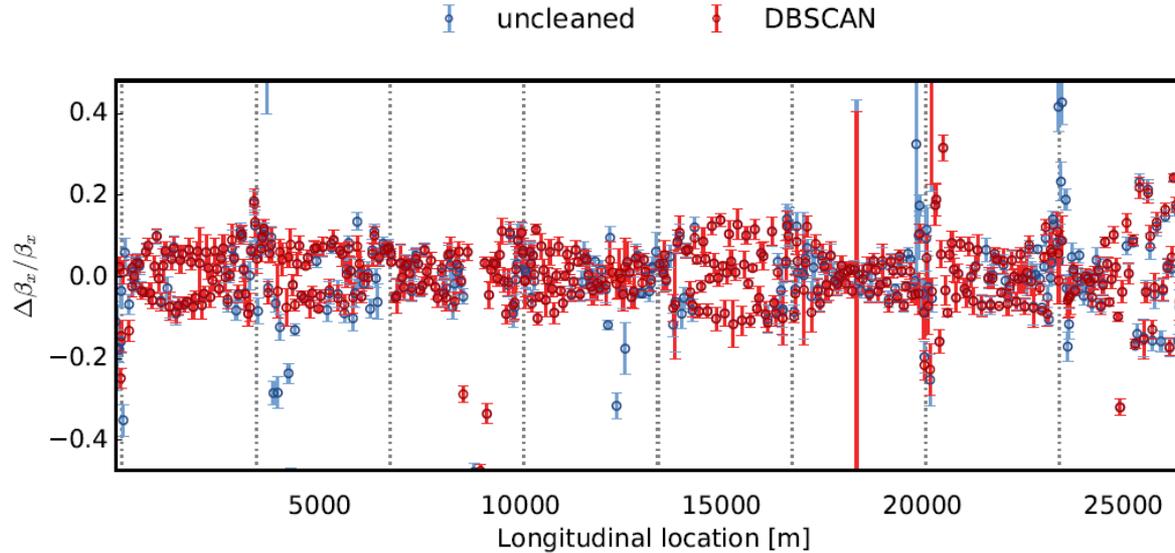
2. Cluster analysis of measured signal

Conclusions:

- DBSCAN is more suitable for the given problem compared to K-means



2. Cluster analysis of measured signal



- the cluster constellation is stable over several runs
- more information might be obtained from analysing the clusters of good BPMs

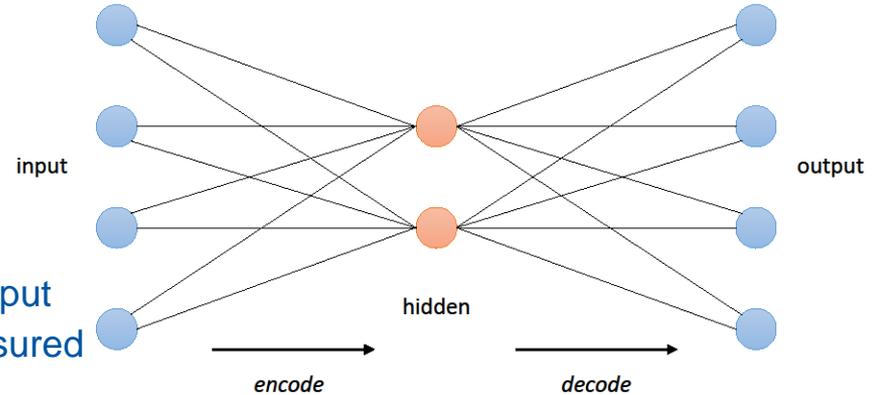
3. Autoencoder for anomaly detection

How does an autoencoder work?*

- Tries to reproduce in its output whatever comes in the input $\rightarrow f(x) = x$
- Encoder: compressing the input data to lower dimensions
Decoder: Reconstructing the data into original input
- The difference between output and input is measured using a loss function

General idea of bad BPMs detection:

- Since the majority of BPMs are good, the trained model will learn the behaviour of valid signal
- The loss function will measure the difference between a given point and the learned general case
- Loss above a defined threshold \rightarrow anomaly detected!



* <https://cds.cern.ch/record/2274402>

4. Conclusions and future tasks

- **Cluster analysis**
 - Cluster analysis improvements potentially possible using ensemble methods (e.g. Random Forest*)
 - The results of cluster analysis can be used as trainings data for a binary classifier to detect bad BPMs
 - Classifier model that outputs the probability for a BPM to be bad
- **Deep neural networks**
 - Building a deep autoencoder no feature engineering will be needed since the hidden layers will extract and learn the relevant features → direct application on turn-by-turn data is possible
- **Beyond anomaly detection**
 - Study on prediction based optics correction using neural networks

http://scikit-learn.org/stable/modules/outlier_detection.html#isolation-forest



www.cern.ch